## Remarks

By this paper, independent claims 1, 11, 21 & 22 are amended to more clearly point out and distinctly claim certain aspects of the present amendments. Specifically, the *copying* in the amended independent claims is recited to be *performed by the currently executing, first software module of the system.* This copying by the currently executing, first software module comprises copying update control code from the first software module to memory space outside a memory location of the first software module. Support for this language can be found throughout the Application as filed. For example, referenced FIG. 3-5, and the discussion thereof beginning at paragraph [20026]. Claims 1-31 remain pending.

In the Office Action, original claims 1-5, 8-15, 18-26 & 29-31 were rejected under 35 U.S.C. §102(e) as being anticipated by Goodman et al (U.S. Patent No. 7,089,547; hereinafter Goodman), while claims 6, 7, 16, 17, 27 & 28 were rejected under 35 U.S.C. §103(a) as being unpatenable over Goodman in view of Farkus et al. (U.S. Patent No. 7,099,967; hereinafter Farkus). These rejections are respectfully traversed to any extent deemed applicable to the claims presented herewith, in reconsideration thereof is requested for the reasons set forth below.

As set forth in the independent claims presented, Applicant's invention is a non-disruptive method for replacing a first software module of a system with a second software module. The method includes:

- copying *by a currently executing, first software module of the system, update control code from the first software module* to memory space outside a memory location of the first software module (see Fig 3);
- replacing the currently executing, first software module with a second software module by storing the second software module in memory at a location which at least partially overlies the first software module, wherein the replacing includes employing the update control code copied from the first software module (see FIG. 4); and
- beginning execution of the second software module *without resetting the system.*

Applicant respectfully submits that at least the above-italicized aspects of their recited protocol patentably distinguish their invention from the teachings and suggestions of Goodman and Farkus, either alone or in combination.

With respect to Applicant's initially recited copying protocol, Figs. 2, 3 & 5 of Goodman, as well as column 4, lines 63-67 & column 5, lines 1-10 thereof are cited in the Office Action. FIGS. 2 & 3 of Goodman illustrate a boot sector, code image 1 and code image 2 (FIG. 2) and update firmware code images (FIG. 3). FIG. 5 is a flowchart depicting selection of a firmware code image for execution *after power on or reset 501.* Column 4, lines 63 and Column 5, line 10 of Goodman teach:

> Both copies, or images, of the operational code are independently executable. In some processors, position independent code may be supported, which employs relative addressing. Thus, either copy is executable in that the same relative addresses are employed for either copy. However, some processors and compliers do not support position independent code, preventing execution of more than one copy of operational code. As discussed above, additional memory may be used to copy either of the two code images into a RAM or other memory area for execution, and the firmware would be compiled to run at the address of the newly copied code in RAM. However, existing embedded systems may not have additional memory to hold a copy of the code image, and new systems would have to incur the additional cost and board space of the copy memory.

Applicant respectfully submits that a careful reading of the above-noted teaching of Goodman fails to uncover any ~~teaching or~~ suggestion of Applicant's copying protocol. In Applicant's recited invention, *the copying is by a currently executing, first software module of a system.* In the process flow of FIG. 5 in Goodman, boot sector 201 is responsible for the illustrated steps. The boot sector 201 is separate from code image 1 or code image 2 as illustrated in FIG 2. Thus, there is no copying in Goodman by a currently executing, first software module of a system, update control code from that first software module. Again, this aspect of Applicant's invention is depicted in FIG. 3 of the present application, wherein firmware update control code is copied from the illustrated first software module to be replaced to a memory location outside of a memory location of the first software module. For at least this reason, Applicant respectfully submits that the independent claims presented herewith patenably distinguish over Goodman.

In addition, Applicant's independent claims recite replacing the currently executing, first software module with the second software module by storing the second software module in memory at a location which at least partly overlies the first software module. The replacing employees the update control code *copied from the first software module.* Goodman, column 3, lines 23-24 are cited. These lines describe replacing the firmware code image to be updated with an updated firmware code image. However, Applicant respectfully submits that these lines of Goodman, as well as the balance of the patent, do not describe *replacing a currently executing, first software module with a second software module, nor does Goodman teach that the replacing employs update control code copied from the currently executing, first software module of the system.* FIG. 4 of the Application illustrates the use of the firmware update control code and the copying of the new module (second software module) over the old module being overlaid (first software module). No similar functionality is taught or suggested by Goodman. For at least this additional reason, Applicant respectfully submits that the independent claims presented herewith patentably distinguish over the implied art.

Still further, Applicant's independent claims recite beginning execution of the second software module *without resetting the system.* Cited against this protocol is column 7, lines 34-55 of Goodman. These lines describe the process flow of FIG. 4 in Goodman for updating firmware of the system. However, the cited lines of Goodman actually teach away from Applicant's recited invention. Column 7, lines 53-56 state: "The firmware update process ends at step 437 where the embedded system may be *reset* to begin execution of the new update code image." (emphasis added.) This is clearly contrary to Applicant's recited independent claims, wherein execution of the second software module occurs *without resetting the system.* Since Goodman expressly teaches resetting the system prior to beginning execution of the new update code image, there is no teaching therein of Applicant's recited invention. For this additional reason, Applicant respectfully submits that the independent claims presented patentably distinguish over Goodman.

For at least the above-noted reasons, the independent claims presented herewith are believed patentable over the applied art. Reconsideration and withdrawal of the rejection thereto are therefore respectfully requested. The dependent claims are believed allowable for the same reasons as the independent claims, as well as for their own additional characterizations.

For example, dependent claims 2, 12 & 23 recite that the replacing of the currently executing, first software module includes *executing the update control code copied from the first software module* during the replacing of the first software module with the second software module. As noted above, the update control code is initially copied by the currently executing, first software module to memory space outside a memory location of the first software module, and is then executed during the replacing of the first software module with the second software module. A careful reading of Goodman fails to uncover any teaching or suggestion of this concept. Goodman does describe replacing one code image with another code image, but the protocol for replacing the image is distinct from that recited Applicant's claims. There is no copying by a currently executing, first software module of the system update control code from that first software module to memory space outside a memory location of the first software module, or subsequent execution of that update control code that was copied from the first software module during the replacing of the first software module with the second software module. Thus, Applicant's protocol recited in these claims is believed patentable over the applied art, and withdrawal of the rejection thereof is respectfully requested.

Dependent claims 5, 15 & 26 recite that the update control code further includes control code for branching to an entry point of the second software module upon completion of the replacing to facilitate the beginning execution of the second software module (again, without resetting the system as recited in the independent claims). The Office Action does not appear to address the substance of these claims. Further, a careful reading of Goodman fails to uncover any discussion of update control code being copied from a currently executing, first software module of the system to memory space outside the first software module, or the subsequent execution of that update control code to facilitate replacement of the first software module with the second software module, or the branching from the update control code to an entry point of the second software module upon completion of the replacing to begin execution of the second software module *without resetting the system.* Thus, Applicant respectfully submits that these claims patentably distinguish over the applied art.

Dependent claims 10, 20 & 31 further recite that the replacing includes employing a hardware based direct memory access (DMA) operation to save the second software module to a target memory space *and wherein the copying update control code includes copying the update* *control code to memory space outside the target memory space, and wherein the update control code includes control code for determining when the DMA operation is completed and for branching to an entry point of the second software module upon completion of the DMA operation.* The Goodman summary does not address the above-italicized functionality of Applicant's dependent claims 10, 20 & 31. The functions recited in Goodman for which the DMA is employed are distinct from the protocol recited in these dependent claims at issue. As noted above, Goodman teaches resetting the system after replacing a first image with a second image before beginning execution of the new image. This is clearly distinct from Applicant's recited protocol wherein the update control code includes an operation for branching to an entry point of the second software module upon completion of the recited DMA operation. Thus, these claims are believed patentable over the applied art.

Farkus is cited in the Office Action in connection with subject matter recited in Applicant's dependent claims 6, 7, 16, 17, 27 & 28. Without acquiescing to the characterizations of Farkus stated in the Office Action, Applicant respectfully submits that a careful reading thereof fails to uncover any teaching or suggestion of the above-noted deficiencies of Goodman when applied against the independent claims presented.

All claims are believed to be in condition for allowance, and such action is respectfully requested.

*Should any issue remain unresolved, however, Applicant's undersigned representative requests a telephone interview with the Examiner to further discuss the matter in the hope of advancing prosecution of the subject application.*

Respectfully submitted,

Kevin P. Radigan

Kevin P. Radigan
Attorney for Applicant
Registration No.: 31,789

Dated: December 14, 2007.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579